

**ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД УКООПСПІЛКИ  
«ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ»**

**НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ БІЗНЕСУ ТА СУЧАСНИХ  
ТЕХНОЛОГІЙ**

**ФОРМА НАВЧАННЯ ДЕННА  
КАФЕДРА МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ ТА СОЦІАЛЬНОЇ  
ІНФОРМАТИКИ**

**Допускається до захисту**  
Завідувач кафедри \_\_\_\_\_ О.О. Ємець  
(підпис)  
« \_\_\_\_\_ » \_\_\_\_\_ 2021 р.

**ПОЯСНЮВАЛЬНА ЗАПИСКА  
ДО БАКАЛАВРСЬКОЇ РОБОТИ**

**на тему**

**АЛГОРИТМІЗАЦІЯ ТА ПРОГРАМУВАННЯ ТРЕНАЖЕРА  
«КЛАСИФІКАЦІЯ ГРАМАТИК, ПРИКЛАДИ ВИКОРИСТАННЯ»  
ДИСТАНЦІЙНОГО НАВЧАЛЬНОГО КУРСУ «ТЕОРІЯ  
ПРОГРАМУВАННЯ»**

**зі спеціальності 122 «Комп'ютерні науки»**

**Виконавець роботи** Любовецький Тарас Олегович

\_\_\_\_\_ « \_\_\_\_\_ » \_\_\_\_\_ 2021р.  
(підпис)

**Науковий керівник** доц., каф.-мат. наук, Черненко О.О.

\_\_\_\_\_ « \_\_\_\_\_ » \_\_\_\_\_ 2021р.  
(підпис)

**ПОЛТАВА 2021р.**

**ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД УКООПСПІЛКИ  
«ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ»**

**ЗАТВЕРДЖУЮ**

Завідувач кафедри \_\_\_\_\_ **О.О. Ємець**  
« 8 » вересня 2020р.

**Завдання та календарний графік  
виконання дипломної роботи**

**Студент(ка) спеціальності 122 «Комп'ютерні науки»**  
**Прізвище, ім'я, по батькові Любовецький Тарас Олегович**

1. Тема **Алгоритмізація та програмування тренажера «Класифікація граматик, приклади використання» дистанційного навчального курсу «Теорія програмування»** затверджена наказом ректора № 121-Н від « 1 » вересня 2020 р.  
Термін подання студентом бакалаврської роботи « 20 » травня 2021 р.
2. Вихідні дані до дипломної роботи: публікації з теми навчальні тренажери в дистанційних курсах з комп'ютерних наук.
3. Зміст пояснювальної записки (перелік питань, які потрібно розробити)

***ВСТУП***

***РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ***

***РОЗДІЛ 2. ІНФОРМАЦІЙНИЙ ОГЛЯД***

- 2.1. Здійснення оперативного контролю за навчальною діяльністю*
- 2.2. Приклад тестових завдань в тренажерах*

***РОЗДІЛ 3. ТЕОРЕТИЧНА ЧАСТИНА***

- 3.1. Класифікація граматик, приклади використання*
- 3.2. Алгоритмізація задачі за темою роботи*
- 3.3 Блок-схема алгоритму тренажера*

***РОЗДІЛ 4. ПРАКТИЧНА ЧАСТИНА***

- 4.1 Опис створення тренажера*
- 4.2 Інструкція для роботи з тренажером*

***ВИСНОВКИ***

***СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ***

***ДОДАТОК А***

4. Перелік графічного матеріалу: 3-4 аркуші блок-схем, інші необхідні ілюстрації.

### 5. Консультанти розділів бакалаврської роботи

Розділ	Прізвище, ініціали, посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1. Постанова задачі	Черненко О.О.	8.09.20	8.09.20
2. Інформаційний огляд	Черненко О.О.	8.09.20	8.09.20
3. Теоретична частина	Черненко О.О.	8.09.20	8.09.20
4. Практична реалізація	Черненко О.О.	8.09.20	8.09.20

### 6. Календарний графік виконання бакалаврської роботи

Зміст роботи	Термін виконання	Фактичне виконання
1. Вступ	10.05.21	
2. Вивчення методичних рекомендацій та стандартів та звіт керівнику	15.09.20	
3. Постановка задачі	1.10.20	
4. Інформаційний огляд джерел бібліотек та інтернету	2.11.20	
5. Теоретична частина	1.02.21	
6. Практична частина	17.05.21	
7. Закінчення оформлення	21.05.21	
8. Доповідь студента на кафедрі	28.05.21	
9. Доробка (за необхідністю), рецензування	14.06.21	

Дата видачі завдання « 8 » вересня 2020 р.

Студент Любовецький Тарас Олегович

Науковий керівник \_\_\_\_\_ доц., каф.-мат. наук, Черненко О.О.  
(підпис)

### **Результати захисту бакалаврської роботи**

Дипломна робота оцінена на \_\_\_\_\_  
(балів, оцінка за національною шкалою,

оцінка за ECTS)

Протокол засідання ЕК № \_\_\_\_\_ від « \_\_\_\_\_ » \_\_\_\_\_ 2021 р.

Секретар ЕК \_\_\_\_\_  
(підпис) (ініціали та прізвище)

## РЕФЕРАТ

**Записка:** 46 стор., в т.ч. основна частина 43 стор., джерел – 10.

**Мета бакалаврської роботи** – алгоритмізація та програмування елементів тренажера «Класифікація граматик, приклади використання» дистанційного навчального курсу «Теорія програмування».

**Об'єктом розробки** є процес дистанційного навчання математичним дисциплінам.

**Предмет розробки** — алгоритм роботи тренажеру з теми «Класифікація граматик, приклади використання».

**Методи, які були використані для розв'язування задачі** – програмна реалізація була здійснена в середовищі розробки IDE Visual Studio та за допомогою платформи Unity.

## ЗМІСТ

ВСТУП.....	4
РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ.....	6
РОЗДІЛ 2. ІНФОРМАЦІЙНИЙ ОГЛЯД.....	8
2.1. Здійснення оперативного контролю за навчальною діяльністю.....	8
2.2. Приклад тестових завдань в тренажерах .....	10
РОЗДІЛ 3. ТЕОРЕТИЧНА ЧАСТИНА .....	12
3.1. Класифікація граматик, приклади використання .....	12
3.2. Алгоритмізація задачі за темою роботи .....	17
3.3 Блок-схема алгоритму тренажера.....	26
РОЗДІЛ 4. ПРАКТИЧНА ЧАСТИНА .....	27
4.1 Опис створення тренажера.....	27
4.2 Інструкція для роботи з тренажером.....	33
ВИСНОВКИ.....	42
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ .....	44
ДОДАТОК А.....	45

## ВСТУП

Головне завдання викладача – це пробудження та розвиток інтересу. Інтерес – рушійна сила пізнання та навчання. Інтерес до заняття починається з викладача. Але зараз для розвитку інтересу до навчання на занятті недостатньо лише особистісних якостей викладача. Необхідно створити і нові технічні умови навчання. Тому бажано звернути увагу на використання новітніх інформаційних технологій.

XXI століття – це час переходу до високотехнологічного інформаційного суспільства, в якому якість людського потенціалу, рівень освіченості й культури всього населення набувають вирішального значення.

Комп'ютер сприяє не тільки розвитку самостійності, творчих здібностей учнів, а й дозволяє змінити саму технологію надання освітніх послуг, зробити заняття більш наочним і цікавим. Комп'ютер забезпечує інтенсифікацію діяльності вчителя та учнів на парі, сприяє здійсненню диференціації та індивідуалізації навчання, розвитку спеціальної або загальної обдарованості, формуванню політехнічних знань, посилює міжпредметні зв'язки. Все це дає можливість покращити якість навчання.

Використання інформаційних технологій може відбуватися різними способами, відповідно до потреб, рівня володіння різними програмами та наявності сертифікованих програм в системі вищої освіти.

*Мета бакалаврської роботи* – алгоритмізація та програмування елементів тренажера «Класифікація граматик, приклади використання» дистанційного навчального курсу «Теорія програмування».

*Об'єктом розробки* є процес дистанційного навчання математичним дисциплінам.

*Предмет розробки* — алгоритм роботи тренажеру з теми «Класифікація граматик, приклади використання».

Перелік використаних методів – застосування граматик.

Бакалаврська робота складається з чотирьох розділів. У першому розділі розглянуто постановку задачі. У другому розділі описано здійснення оперативного контролю за навчальною діяльністю, приклад тестових завдань в тренажерах. У третьому розділі представлено класифікацію граматик, приклади використання, алгоритмізацію задачі за темою роботи. У четвертому розділі представлено програмну реалізацію тренажера.

У додатку міститься диск з матеріалами проекту.

Обсяг пояснювальної записки: 46 стор., в т.ч. основна частина - 43 стор., додатки - 1 стор., джерела - 10 назв.

## РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ

Тренажерне прикладне програмне забезпечення – це програмний педагогічний засіб, який симулює певну реальну або навчальну ситуацію. Слід також розрізняти таких два поняття як модель та симулятор.

Модель – це представлення реального об'єкта або явища за допомогою певної формалізованої мови описання, що відтворює, імітує, відображає принципи внутрішньої організації або функціонування, певні властивості, ознаки чи характеристики об'єкта дослідження – математична модель.

Симулятор – це більш наближений до реальності спосіб представлення об'єкту або процесу за допомогою графічних, анімаційних та мультимедійних засобів.

Головними завданнями або ж забезпеченням програм-тренажерів є:

- послідовне виведення на екран завдань заданої складності з вибраної теми;
- контроль за діями користувача з розв'язання запропонованого завдання;
- миттєву реакцію на неправильні дії;
- виправлення помилок користувача;
- демонстрацію правильного розв'язання завдання;
- виведення підсумкового повідомлення про результати роботи користувача (можливо, з рекомендаціями чи порадами).

Комп'ютерні тренажери були і є незамінними при підготовці майбутніх спеціалістів. Дана практика поширилася і на навчальний процес, яка дала змогу формувати практичні уміння і навички при вивченні навчальних предметів та застосовувати набуті знання на практиці, оскільки вивчення дисциплін також невід'ємне від тренування: студент має впевнено розв'язувати задачі і вправи, виконувати завдання різного характеру, набувати навичок безпомилкового письма і здійснення відповідної діяльності на рівні автоматизму.



Оцінювання виконання завдань та вправ на тренажерах здійснюється автоматично, тому вчителю не треба витрачати час для того, щоб перевірити завдання. Програми-тренажери можна використовувати для різних типів уроку та на будь-якому етапі уроку. Крім того, студенти можуть самостійно використовувати їх для самопідготовки та самонавчання. Це є ефективний засіб та метод для поліпшення навичок усіх студентів.

Програма-тренажер повинна містити:

- стартову сторінку;
- сторінки з питаннями;
- повідомлення про помилку;
- сторінку з результатами проходження тренажера.

Для цього потрібно реалізувати наступні функції:

- перехід між основними панелями:
  - стартова сторінка;
  - сторінка з питаннями;
  - сторінка з результатами;
- перехід до наступного питання;
- перехід до попереднього питання;
- перевірка вибраної відповіді на питання;
- виведення повідомлення про помилку;
- підрахунок результатів проходження тренажера.

## РОЗДІЛ 2. ІНФОРМАЦІЙНИЙ ОГЛЯД

### 2.1. Здійснення оперативного контролю за навчальною діяльністю

Здійснення оперативного контролю за навчальною діяльністю студентів є однією з найважливіших проблем дистанційного навчання. Рішення проблеми контролю якості дистанційного навчання, його відповідності освітнім стандартам має принципове значення для успіху всієї системи дистанційного навчання. До основних переваг контролю знань в системах дистанційного навчання можна віднести наступні:

- Об'єктивність. Виключається чинник суб'єктивного підходу з боку екзаменатора. Обробка результатів проводиться через комп'ютер;
- Демократичність. Всі студенти знаходяться в рівних умовах;
- Індивідуальність. З'являється можливість кожному студенту вибрати час проведення контролю, кількість виконання завдань під час одного сеансу роботи і т.п.

Контроль якості буває різних видів і форм, а також може здійснюватися за допомогою різноманітних методів. Методи контролю якості навчальної діяльності повинні забезпечувати систематичне, повне, точне і оперативне отримання інформації про навчальний процес. Контроль, як правило проводиться з дотриманням наступних вимог:

- систематичність, регулярність проведення контролю на всіх етапах освітнього процесу;
- індивідуальний характер контролю, що вимагає здійснення контролю за роботою кожного студента в залежності від його індивідуальної освітньої траєкторії;
- різноманітність форм проведення, що забезпечує підвищення інтересу студентів до його проведення і результатами;

- всебічність, яка полягає в тому, що контроль повинен охоплювати всі розділи навчальної програми, забезпечувати перевірку теоретичних знань, практичних умінь і компетенцій студентів;

- об'єктивність контролю, що виключає навмисні, суб'єктивні і помилкові оціночні судження викладача-тьютора, засновані на недостатньому вивченні студентів, психолого-педагогічних особливостей їх психофізичного розвитку. Дотримання зазначених вимог забезпечує надійність контролю та виконання ним своїх завдань у процесі дистанційного навчання.

Безумовно, перед тим, як сформувати систему поточного контролю, формується організаційна структура матеріалу, що вивчається.

Важливе значення при впровадженні дистанційного навчання математичних дисциплін має розробка і програмне виконання віртуальних тренажерів. Це засіб навчання дозволяє підвищити ефективність засвоєння знань, якість розуміння матеріалу, виробити професійно-орієнтовані уміння, навички в дослідженні властивостей різноманітних технічних та фізичних процесів. Так, тренажер дозволяє створювати об'єкти і образи неіснуючі в реальності, виконувати віртуальні дії.

Тренажери дозволяють представити матеріал в різному вигляді, контролювати отримані знання у вигляді гри, представити предметну область на різних рівнях глибини засвоєння матеріалу і детальності інформації. З їх допомогою можна отримати навички розв'язання типових практичних завдань, використовувати бази навчальних матеріалів: електронні каталоги, бібліотеки ілюстрацій, глосарії тощо.

Віртуальний тренажер дозволяє реалізувати наступні аспекти в навчанні:

- врахування рівня підготовки, мотиву навчання;
- орієнтація на індивідуальні особливості студента;
- нормування обсягу наданого матеріалу;
- завдання алгоритму виконання роботи.

Разом з тим, впровадження таких засобів навчання пред'являє високі вимоги до викладача дисципліни на етапі розробки та налагодження електронних

тренажерів. При цьому викладач повинен знати ці можливості, вміти формувати навчальний матеріал з різних видів, представляти якісний сценарій тренажера, розробити ефективну шкалу оцінки. Ці електронні засоби надають неоціненну допомогу як студентам (вони дозволяють більш наочно піднести матеріал, повторювати вправи до повного розуміння і закріплення досліджуваного матеріалу), так і викладачеві (не потрібно його постійної присутності, тренажер неупереджено оцінюють отримане рішення, методичні рекомендації для кожного кроку видаються автоматично, програма сама вказує на допущені помилки). Реалізація тренажерів в рамках досліджуваного теоретичного і практичного матеріалу дозволяє сформувати і закріпити практичні вміння та навички у виконанні математичних дій [2].

## **2.2. Приклад тестових завдань в тренажерах**

Ще однією формою організації практичної діяльності є *тестові завдання* (рисунки 2.1). Серед них тести, що потребують вибору вірної відповіді (рисунки 2.2), тести з необхідністю внесення результату за допомогою клавіатури, тести складання яких відбувається в декілька етапів (рисунки 2.3), тести на відповідність, тести на порядок дій.

До переваг тестового контролю знань можна віднести чіткість та швидкість виконання, об'єктивність оцінювання, можливість вимірювати результати навчання. Недоліки тестових завдань містяться у тому, що викладач не може контролювати процес розв'язування, з'являється можливість вгадування результатів, контролюються тільки той рівень знань, що закладений в тестах [3].

Шаг 1 Шаг 2 Шаг 3 Шаг 4 Шаг 5 Шаг 6 Шаг 7 Вывод

Проверьте данную функцию на чётность.

$$y = 4x^3 + 8x^2 - 4x - 3$$

$y(-x) =$

Выполняется ли одно из условий:

(1)  $y(-x) = y(x)$

(2)  $y(-x) = -y(x)$

Значит, данная функция является

Калькулятор

Рисунок 2.1 – Віртуальний тренажер «Характеристики функцій».

Знайти похідну функції  $y = \sqrt{x^2 - 3} + 2x$

☐  $\frac{2x}{\sqrt{x^2 - 3}}$   
☐  $\sqrt{2x - 3}$   
☐  $\frac{1}{\sqrt{2x}}$   
☐  $\frac{x}{\sqrt{x^2 - 3}} + 2$

Рисунок 2.2 – Тест із варіантами відповідей за темою «Похідна».

Задан ряд  $\sum_{n=1}^{\infty} \left(\frac{9n}{n+1}\right)^{2n}$ .

Укажите признак по которому проведено исследование ряда на сходимость и сделайте вывод о сходимости заданного ряда.

Проведено исследование ряда на сходимость по признаку

$$K = \lim_{n \rightarrow \infty} \sqrt[n]{\left(\frac{9n}{n+1}\right)^{2n}} = \lim_{n \rightarrow \infty} \left(\frac{9n}{n+1}\right)^{\frac{2n}{n}} = \lim_{n \rightarrow \infty} \left(\frac{9n}{n+1}\right)^2 = \left(\frac{9}{1}\right)^2 > 1$$

Заданный ряд

Рисунок 2.3 – Розширений тест з теми «Ряди».

## РОЗДІЛ 3. ТЕОРЕТИЧНА ЧАСТИНА

### 3.1. Класифікація граматик, приклади використання

Хомський визначив 4 класи граматик. Граматики 0-го типу, **загального вигляду**, або рекурсивно-перерелічувані визначаються як граматики, що відповідають нашому визначенню без обмежень на типи продукцій. Це найбільш загальний клас, інші граматики можуть бути отримані накладенням обмежень на продукції граматик 0-го типу. Граматики 0-го типу еквівалентні машинам Тьюринга.

Перше обмеження: задати, щоб для всіх продукцій  $\alpha \rightarrow \beta$  довжина рядка  $\alpha$ , обчислена в кількості символів, була не більше довжини рядка  $\beta$ . Граматики, усі продукції яких задовольняють дане обмеження, називаються граматиками 1-го типу, або **контекстозалежними**. Граматики 1-го типу еквівалентні лінійно-обмеженим автоматам.

Якщо, крім уже названого обмеження, у лівій частині продукції повинен знаходитися тільки один нетермінал, граматика називається граматикою 2-го типу або контекстовільною граматикою. Приклади  $G_1, G_2$  представляють 2-й тип. У контекстовільних граматиках зручно дозволити продукцію  $S \rightarrow \varepsilon$ , хоча, строго кажучи, вона не дозволена навіть у контекстозалежних граматиках. Граматики 2-го типу еквівалентні магазинним автоматам (push-down).

Останній клас граматик – граматики 3-го типу, або **регулярні** (автоматні) граматики. Мова такої граматики називається регулярною. Регулярну мову можна визначити регулярним виразом і навпаки. Регулярні мови і регулярні вирази еквівалентні скінченним автоматам.

Ієрархія Хомського є включаючою (рис 3.1).

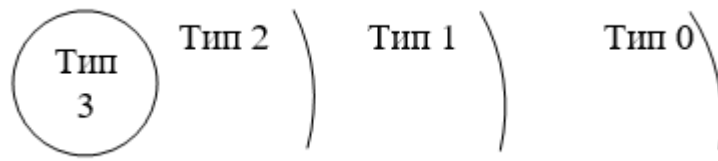


Рисунок 3.1 - Ієрархія Хомського для граматик

Формальні мови класифікуються відповідно до типів граматик, якими вони задаються. Однак та сама мова може бути задана різними граматиками, що належать до різних типів. У такому випадку вважається, що мова належить до найбільш простого з них. Так, мова, описана граматикою з фразовою структурою, контекстозалежною і контекстовільною граматиками, буде контекстовільною.

Найбільш складні – мови загального вигляду (сюди можна віднести природні мови), найпростіші – регулярні мови.

Розглянемо поняття граматика. **Граматика** має чотири компоненти.:

1. Безліч термінальних символів, іноді іменованих токенів. Термінали представляють собою елементарні символи мови, з яких формуються рядки.

2. Безліч нетерміналів, які іноді називають синтаксичними змінними. Кожен нетермінал – це безліч рядків терміналів.

3. Безліч продукцій, кожна з яких складається з нетермінала, який називають заголовком або лівою частиною продукції, стрілки і послідовності терміналів і / або нетерміналів, які називають тілом або правою частиною продукції. Інтуїтивно призначення продукції - визначити один з можливих видів конструкції; якщо заголовний нетермінал представляє конструкцію, то тіло являє записуваний вигляд конструкції.

4. Один з нетермінальних символів, що вказується як стартовий або початковий.

Формально **граматика** визначається як наступна четвірка компонентів  $(V_T, V_N, P, S)$ .

Тут:

$V_T$  – алфавіт термінальних символів або терміналів;

$V_N$  – алфавіт нетермінальних символів або нетерміналів,  $V_T \cap V_N = \emptyset$ ;

$P$  – множина продукцій (або правил) вигляду  $\alpha \rightarrow \beta$ ,  $\alpha$  складається з одного або більше символів  $V$ ,  $\beta$  – з нуля або більше символів  $V$ , де  $V = V_T \cup V_N$ ;

$S$  – стартовий символ (або аксіома).

### Приклади.

1. Мова  $\{x^n y^n \mid n > 0\}$  описується граматикою

$$G_1 = (\{x, y\}, \{S\}, P, S).$$

Тут  $P = \{S \rightarrow xSy, S \rightarrow xy\}$ .

2. Граматикою для мови  $\{x^m y^n \mid m, n \geq 0\} \in G_2 = (\{x, y\}, \{S, B\}, P, S)$ .

Тут набір продукцій  $P$  має вигляд

$$\begin{array}{ll} S \rightarrow xS, & S \rightarrow y, \\ S \rightarrow yB, & B \rightarrow yB, \\ S \rightarrow x, & B \rightarrow y. \end{array}$$

Оскільки порожній рядок також належить мові, у набір  $P$  також входить продукція  $S \rightarrow \varepsilon$ .

Рядок  $xxyy$  генерується в такий спосіб:

$$S \Rightarrow xS \Rightarrow xxS \Rightarrow xxyB \Rightarrow xxyyB \Rightarrow xxyy.$$

Нетермінали записуються словами в дужках  $\langle \rangle$  або великими латинськими буквами. Термінали за необхідності часом беруться в апострофи. Як і в мові БНФ, замість продукцій вигляду  $v \rightarrow w_1 w w_2$  і  $v \rightarrow w_1 w_2$  записується продукція  $v \rightarrow w_1 [w] w_2$ , а замість продукцій  $v \rightarrow w_1 u_1 w_2$  і  $v \rightarrow w_1 u_2 w_2$  – продукція  $v \rightarrow w_1 (u_1 \mid u_2) w_2$ .

3. Множину продукцій граматки  $G_1 = (\{a, 1, 2\}, \{A, B, C, D\}, \{A \rightarrow BC, A \rightarrow BD, A \rightarrow B, B \rightarrow a, C \rightarrow 1, D \rightarrow 2\}, A)$  можна переписати у вигляді  $\{A \rightarrow B[C \mid D], B \rightarrow a, C \rightarrow 1, D \rightarrow 2\}$ .



Як бачимо, продукції граматики дуже схожі на БНФ (форма Бекуса-Наура) як за формою, так і за змістом – лише замість знака "::=" вживається " $\Rightarrow$ ". Проте в лівій частині їх продукцій може бути не поодиноким нетермінал, а цілий ланцюжок, у якому обов'язково є нетермінал. За рахунок такого узагальнення граматики виявляються більш потужним засобом задання мов, ніж системи БНФ, тобто існують мови, які задаються граматиками, але не задаються БНФ. Тепер опишемо спосіб, у який граматика  $G = (V_T, V_N, P, S)$  задає мову.

1. На множині слів об'єднаного алфавіту  $(V)^*$  означається **відношення безпосередньої виводимості**, позначене знаком  $\Rightarrow_G$  (або  $\Rightarrow$ , коли відомо, якою саме є  $G$ ):  $v \Rightarrow_G w$ , якщо  $v = x_1 u x_2$ ,  $w = x_1 u x_2$ ,  $u \rightarrow y \in P$ .

При цьому кажуть, що  $w$  **безпосередньо виводиться** з  $v$  **застосуванням продукції**  $u \rightarrow y$ . Наприклад, у граматиці  $G_1$  із прикладу 3  $BC \Rightarrow aC$  застосуванням продукції  $B \rightarrow a$ ,  $aC \Rightarrow a1$  застосуванням  $C \rightarrow 1$ .

2. На множині  $(V)^*$  означається **відношення виводимості**, позначене  $\Rightarrow^*_G$  (або  $\Rightarrow^*$ , коли відомо, якою саме є  $G$ ):  $v \Rightarrow^*_G w$ , якщо  $v = w$  або існує послідовність  $w_1, w_2, \dots, w_n$  слів, де  $n \geq 1$ , така, що  $v \Rightarrow w_1, w_1 \Rightarrow w_2, \dots, w_{n-1} \Rightarrow w_n, w_n = w$ . Так, у граматиці  $G_1 BC \Rightarrow^* a1$ ,

оскільки  $BC \Rightarrow aC, aC \Rightarrow a1$ . Послідовність  $v \Rightarrow w_1 \Rightarrow w_2 \Rightarrow \dots \Rightarrow w_n$  називається **виведенням**  $w_n$  із  $v$ , а  $n$  – **довжиною виведення**. Інколи довжину записують замість '\*' таким чином:  $v \Rightarrow^n w$ , наприклад,  $BC \Rightarrow^2 a1$ .

3. Якщо  $S \Rightarrow^*_G w$ , то послідовність  $S \Rightarrow \dots \Rightarrow w$  називається **виведенням слова**  $w$  **у граматиці**  $G$ , а слово  $w$  – **вивідним**. Так, слова .. вивідні в граматиці прикладу 3.

4. Вивідні слова в алфавіті  $X$  називаються **породжуваними**, а множина їх усіх – **мовою, що задається (породжується) граматиною**  $G$ :  $L(G) = \{w \mid w \in X^* \text{ та } S \Rightarrow^* w\}$ .

### Приклади.

4. Граматика  $G_1$  із прикладу 3 задає мову  $\{a, a1, a2\}$ .

### 5. Граматика

$$G_2 = (\{a, \dots, z, 0, \dots, 9\}, \{I, L, D\}, \{I \rightarrow L \mid IL \mid ID, L \rightarrow a \mid \dots \mid z, D \rightarrow 0 \mid \dots \mid 9\}, I)$$

породжує множину ідентифікаторів.

6. Граматика  $G_3 = (\{(\, , )\}, \{S\}, \{S \rightarrow \varepsilon \mid (S)\}, S)$  задає множину "вкладених дужок"  $\{( ^n )^n \mid n \geq 0\}$ .

7. Граматика  $G_4 = (\{a, b, c\}, \{S, A, B, C\}, \{S \rightarrow aSBC \mid abC, CB \rightarrow BC, bB \rightarrow bb, bC \rightarrow bc, cC \rightarrow cc\}, S)$  визначає мову  $\{a^n b^n c^n \mid n \geq 1\}$ .

Граматика називаються **еквівалентними**, якщо задають ту саму мову. Наприклад, граматика  $(\{a, 1, 2\}, \{A\}, \{A \rightarrow a[1 \mid 2]\}, A)$  еквівалентна граматиці з *прикладу 3*, граматика

$$(\{a, \dots, z, 0, \dots, 9\}, \{I, C\}, \{I \rightarrow (a \mid \dots \mid z)C, C \rightarrow \varepsilon \mid C(a \mid \dots \mid z \mid 0 \mid \dots \mid 9)\}, I) -$$

граматиці з *прикладу 5*.

Кожен з рядків, що фігурують у породженні, називається **сентенціальною формою**, а останній рядок, що складається з терміналів, – **сентенцією** (пропозицією) мови.

При цьому

$\Rightarrow$  означає один крок,

$*$   
 $\Rightarrow$  нуль або більше кроків,

$+$   
 $\Rightarrow$  один або більше кроків.

Запис  $\begin{cases} B \rightarrow yB, \\ B \rightarrow y \end{cases}$  можна записувати у більш стислому вигляді:  $B \rightarrow yB \mid y$ .

Ту ж саму мову можна згенерувати багатьма граматиками.

Наше визначення граматики допускає граматики і більш загальних типів, ніж були у прикладах.

Наприклад:  $G_3 = (\{a\}, \{S, N, Q, R\}, P, S)$ , де  $P$  містить продукції:

$$S \rightarrow QNQ,$$

$$Q \rightarrow QR,$$

$$RQ \rightarrow NNQ,$$

...

Тут  $N$  переходить у  $R$ , якщо  $N$  стоїть після  $Q$ , а  $R$  переходить у  $NN$ , якщо воно стоїть перед  $Q$ .

У даному прикладі продукції є контекстозалежними. Контекстнонезалежні продукції мають порожній контекст [4,5].

### 3.2. Алгоритмізація задачі за темою роботи

Після запуску виводиться головне вікно з інформацією, що містить:

- тему;
- інформацію про автора;
- інформацію про керівника;
- перехід до виконання тренажеру;
- вихід.

На кожному кроці виводиться завдання, необхідно вибрати чи вказати відповідь.

**Крок 1.** Відображається завдання: Хомський визначив 4 класи граматик. Граматики 0-го типу, загального вигляду, або:

- рекурсивно-перерелічувані;
- контекстовільні;
- контекстозалежні;
- регулярні.

Якщо відповідь неправильна, то відображається помилка: «Граматики 0-го типу, загального вигляду, або рекурсивно-перерелічувані граматики».

Якщо правильна – відбувається перехід на наступний крок.

**Крок 2.** Відображається завдання: Граматики 0-го типу, загального вигляду, або рекурсивно-перерелічувані визначаються як:

- граматики, усі продукції яких задовольняють дане обмеження: задати, щоб для всіх продукцій  $\alpha \rightarrow \beta$  довжина рядка  $\alpha$ , обчислена в кількості символів, була не більше довжини рядка  $\beta$ ;
- граматики, що відповідають нашому визначенню без обмежень на типи продукцій;
- граматики, в яких, крім уже названого обмеження, у лівій частині продукції повинен знаходитися тільки один нетермінал.

Якщо відповідь неправильна, то відображається помилка: «Граматики 0-го типу, загального вигляду, або рекурсивно-перерелічувані визначаються як граматики, що відповідають нашому визначенню без обмежень на типи продукцій».

Якщо правильна – відбувається перехід на наступний крок.

**Крок 3.** Відображається завдання: Граматики 0-го типу еквівалентні:

- магазинним автоматам (push-down);
- лінійно-обмеженим автоматам;
- скінченим автоматам;
- машинам Тьюринга.

Якщо відповідь неправильна, то відображається помилка: «Граматики 0-го типу еквівалентні машинам Тьюринга».

Якщо правильна – відбувається перехід на наступний крок.

**Крок 4.** Відображається завдання: Хомський визначив 4 класи граматик. Граматики 1-го типу, або:

- рекурсивно-перерелічувані;
- контекстовільні;
- контекстозалежні;
- регулярні.

Якщо відповідь неправильна, то відображається помилка: «Граматики 1-го типу, або контекстозалежні граматики».

Якщо правильна – відбувається перехід на наступний крок.

**Крок 5.** Відображається завдання: Перше обмеження: задати, щоб для всіх продукцій  $\alpha \rightarrow \beta$  довжина рядка  $\alpha$ , обчислена в кількості символів, була не більше довжини рядка  $\beta$ . Граматиками 1-го типу, або контекстозалежними називаються:

- граматики, усі продукції яких задовольняють дане обмеження;
- граматики, що відповідають нашому визначенню без обмежень на типи продукцій;
- граматики, в яких, крім уже названого обмеження, у лівій частині продукції повинен знаходитися тільки один нетермінал.

Якщо відповідь неправильна, то відображається помилка: «Граматиками, усі продукції яких задовольняють дане обмеження, називаються граматиками 1-го типу, або контекстозалежними».

Якщо правильна – відбувається перехід на наступний крок.

**Крок 6.** Відображається завдання: Граматики 1-го типу еквівалентні:

- магазинним автоматам (push-down);
- лінійно-обмеженим автоматам;
- скінченим автоматам;
- машинам Тьюринга.

Якщо відповідь неправильна, то відображається помилка: «Граматиками 1-го типу еквівалентні лінійно-обмеженим автоматам».

Якщо правильна – відбувається перехід на наступний крок.

**Крок 7.** Відображається завдання: Хомський визначив 4 класи граматик. Граматики 2-го типу, або:

- рекурсивно-перерелічувані;
- контекстовільні;
- контекстозалежні;
- регулярні.

Якщо відповідь неправильна, то відображається помилка: «Граматиками 2-го типу, або контекстовільні граматики».

Якщо правильна – відбувається перехід на наступний крок.

**Крок 8.** Відображається завдання: Перше обмеження: задати, щоб для всіх продукцій  $\alpha \rightarrow \beta$  довжина рядка  $\alpha$ , обчислена в кількості символів, була не більше довжини рядка  $\beta$ . Граматикою 2-го типу або контекстовільною граматикою називаються:

- граматика, усі продукції яких задовольняють дане обмеження;
- граматика, що відповідають нашому визначенню без обмежень на типи продукцій;
- граматика, в яких, крім уже названого обмеження, у лівій частині продукції повинен знаходитися тільки один нетермінал.

Якщо відповідь неправильна, то відображається помилка: «Якщо, крім уже названого обмеження, у лівій частині продукції повинен знаходитися тільки один нетермінал, граматика називається граматикою 2-го типу або контекстовільною граматикою.».

Якщо правильна – відбувається перехід на наступний крок.

**Крок 9.** Відображається завдання: Граматики 2-го типу еквівалентні:

- магазинним автоматам (push-down);
- лінійно-обмеженим автоматам;
- скінченим автоматам;
- машинам Тьюринга.

Якщо відповідь неправильна, то відображається помилка: «Граматики 2-го типу еквівалентні магазинним автоматам (push-down)».

Якщо правильна – відбувається перехід на наступний крок.

**Крок 10.** Відображається завдання: Хомський визначив 4 класи граматик. Граматики 3-го типу, або:

- рекурсивно-перерелічувані;
- контекстовільні;
- контекстозалежні;
- регулярні.

Якщо відповідь неправильна, то відображається помилка: «Граматики 3-го типу, або регулярні (автоматні) граматики».

Якщо правильна – відбувається перехід на наступний крок.

**Крок 11.** Відображається завдання: Мова такої граматики називається регулярною. Регулярну мову можна визначити:

- регулярним виразом;
- контекстовільною граматиною;
- регулярним виразом і навпаки.

Якщо відповідь неправильна, то відображається помилка: «Регулярну мову можна визначити регулярним виразом і навпаки».

Якщо правильна – відбувається перехід на наступний крок.

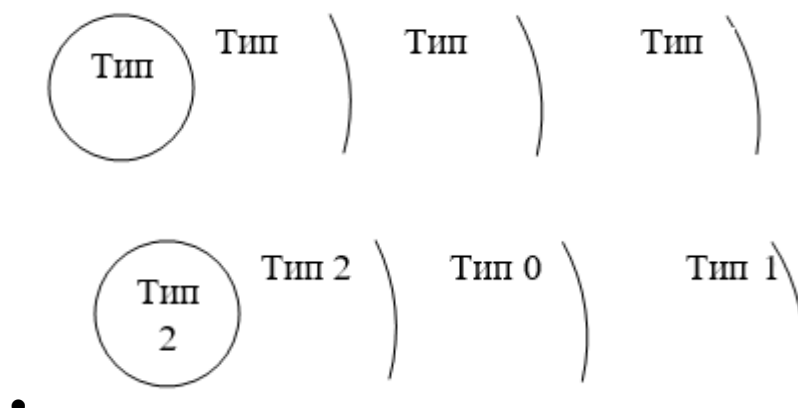
**Крок 12.** Відображається завдання: Граматики 3-го типу еквівалентні:

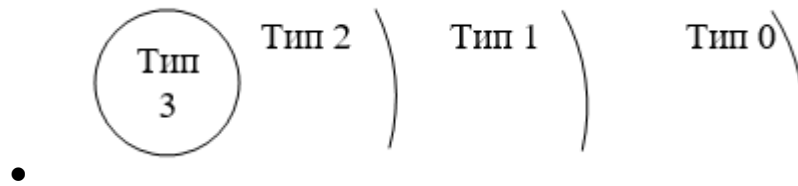
- магазинним автоматам (push-down);
- лінійно-обмеженим автоматам;
- скінченим автоматам;
- машинам Тьюринга.

Якщо відповідь неправильна, то відображається помилка: «Регулярні мови і регулярні вирази еквівалентні скінченим автоматам».

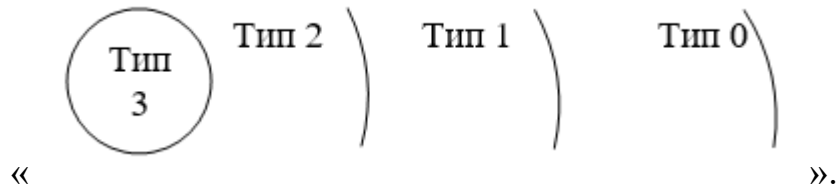
Якщо правильна – відбувається перехід на наступний крок.

**Крок 13.** Відображається завдання: Ієрархія Хомського є включаючою. Вкажіть по зображенню кожен тип:





Якщо відповідь неправильна, то відображається помилка:



Якщо правильна – відбувається перехід на наступний крок.

**Крок 14.** Відображається завдання: Формальні мови класифікуються відповідно до типів граматик, якими вони задаються. Однак та сама мова може бути задана різними граматиками, що належать до різних типів. У такому випадку вважається, що мова належить до найбільш простого з них.

Так, мова, описана граматикою з фразовою структурою, контекстозалежною і контекстовільною граматиками, буде:

- контекстозалежною;
- контекстовільною;
- регулярною.

Якщо відповідь неправильна, то відображається помилка: «Так, мова, описана граматикою з фразовою структурою, контекстозалежною і контекстовільною граматиками, буде контекстовільною».

Якщо правильна – відбувається перехід на наступний крок.

**Крок 15.** Відображається завдання: Приклад 1. Мова  $\{x^n y^n | n > 0\}$  описується граматикою  $G_1 = (\{x, y\}, \{S\}, P, S)$ . Тут  $P$ :

$$P = \{S \rightarrow \_, S \rightarrow \_ \}$$

- $P = \{S \rightarrow xy, S \rightarrow xy\};$
- $P = \{S \rightarrow xSy, S \rightarrow xy\};$
- $P = \{S \rightarrow xS, S \rightarrow xy\}.$

Якщо відповідь неправильна, то відображається помилка: «Тут  $P = \{S \rightarrow xSy, S \rightarrow xy\}$ ».



Якщо правильна – відбувається перехід на наступний крок.

**Крок 16.** Відображається завдання: Приклад 2. Граматикою для мови  $\{x^m y^n | m, n \geq 0\} \in G_2 = (\{x, y\}, \{S, B\}, P, S)$ .

Тут набір продукцій  $P$  має вигляд

$$\begin{array}{ll} S \rightarrow xS, & S \rightarrow y, \\ S \rightarrow yB \quad S \rightarrow yB, & B \rightarrow yB, \\ S \rightarrow x, & B \rightarrow y. \end{array}$$

Оскільки порожній рядок також належить мові, у набір  $P$  також входить продукція  $S \rightarrow \varepsilon$ .

Рядок  $xxuuu$  генерується в такий спосіб:

$$S \Rightarrow \_ \Rightarrow \_ \Rightarrow \_ \Rightarrow \_ \Rightarrow \_$$

- $S \Rightarrow S \Rightarrow xxS \Rightarrow xxyB \Rightarrow xxyuB \Rightarrow xxyuu$
- $S \Rightarrow xS \Rightarrow xxS \Rightarrow xyB \Rightarrow xxyuB \Rightarrow xxyuu$
- $S \Rightarrow xS \Rightarrow xxS \Rightarrow xxyB \Rightarrow xxyuB \Rightarrow xxyuu$

Якщо відповідь неправильна, то відображається помилка: «Рядок  $xxuuu$  генерується в такий спосіб:

$$S \Rightarrow xS \Rightarrow xxS \Rightarrow xxyB \Rightarrow xxyuB \Rightarrow xxyuu$$

Якщо правильна – відбувається перехід на наступний крок.

**Крок 17.** Відображається завдання: Приклад 3. Множину продукцій граматики

$$G_1 = (\{a, 1, 2\}, \{A, B, C, D\}, \{A \rightarrow BC, A \rightarrow BD, A \rightarrow B, B \rightarrow a, C \rightarrow 1, D \rightarrow 2\}, A)$$

можна переписати у вигляді:

$$\{A \rightarrow \_, B \rightarrow \_, C \rightarrow \_, D \rightarrow \_ \}$$

- $\{A \rightarrow B[C|D], B \rightarrow a, C \rightarrow 1, D \rightarrow 2\}$ ».
- $\{A \rightarrow B[C], B \rightarrow a, C \rightarrow 1, D \rightarrow 2\}$ ».
- $\{A \rightarrow B[C|D], B \rightarrow a, C \rightarrow 2, D \rightarrow 3\}$ ».

Якщо відповідь неправильна, то відображається помилка: «Множину продукцій граматики

$$G_1 = (\{a, 1, 2\}, \{A, B, C, D\}, \{A \rightarrow BC, A \rightarrow BD, A \rightarrow B, B \rightarrow a, C \rightarrow 1, D \rightarrow 2\}, A)$$

можна переписати у вигляді  $\{A \rightarrow B[C|D], B \rightarrow a, C \rightarrow 1, D \rightarrow 2\}$ ».

Якщо правильна – відбувається перехід на наступний крок.

**Крок 18.** Відображається завдання: Приклад 4. Граматика  $G_1 = (\{a, 1, 2\}, \{A, B, C, D\}, \{A \rightarrow BC, A \rightarrow BD, A \rightarrow B, B \rightarrow a, C \rightarrow 1, D \rightarrow 2\}, A)$

задає мову:

$\{\_\_\_\_\_\}$

- $\{a1, a2\};$
- $\{a, a, a2\};$
- $\{a, a1, a2\}.$

Якщо відповідь неправильна, то відображається помилка: «Граматика  $G_1$  задає мову  $\{a, a1, a2\}$ ».

Якщо правильна – відбувається перехід на наступний крок.

**Крок 19.** Відображається завдання: Приклад 5. Граматика  $G_2 = (\{a, \dots, z, 0, \dots, 9\}, \{I, L, D\}, \{I \rightarrow L|IL|ID, L \rightarrow a|\dots|z, D \rightarrow 0|\dots|9\}, I)$  породжує:

- множину ідентифікаторів;
- мову;
- множину "вкладених дужок".

Якщо відповідь неправильна, то відображається помилка: «Граматика  $G_2 = (\{a, \dots, z, 0, \dots, 9\}, \{I, L, D\}, \{I \rightarrow L|IL|ID, L \rightarrow a|\dots|z, D \rightarrow 0|\dots|9\}, I)$  породжує множину ідентифікаторів».

Якщо правильна – відбувається перехід на наступний крок.

**Крок 20.** Відображається завдання: Приклад 6. Граматика  $G_3 = (\{(\,,\,)\}, \{S\}, \{S \rightarrow \varepsilon|(S)\}, S))$  задає:

- множину ідентифікаторів;
- мову;
- множину "вкладених дужок"  $\{(^\textit{n})^\textit{n} | n \geq 0\}.$

Якщо відповідь неправильна, то відображається помилка: «Граматика  $G_3 = (\{(\,,\,)\}, \{S\}, \{S \rightarrow \varepsilon|(S)\}, S))$  задає множину "вкладених дужок"  $\{(^\textit{n})^\textit{n} | n \geq 0\}$ ».

Якщо правильна – відбувається перехід на наступний крок.

**Крок 21.** Відображається завдання: Приклад 7. Граматика  $G_4 = (\{a, b, c\}, \{S, A, B, C\}, \{S \rightarrow aSBC|abC, CB \rightarrow BC, bB \rightarrow bb, bC \rightarrow bc, cC \rightarrow cc\}, S)$  визначає:

- множину ідентифікаторів;
- мову  $\{a^n b^n c^n | n \geq 1\}$ ;
- множину "вкладених дужок"  $\{(\text{"})"}^n | n \geq 0\}$ .

Якщо відповідь неправильна, то відображається помилка: «Граматика  $G_4 = (\{a, b, c\}, \{S, A, B, C\}, \{S \rightarrow aSBC|abC, CB \rightarrow BC, bB \rightarrow bb, bC \rightarrow bc, cC \rightarrow cc\}, S)$  визначає мову  $\{a^n b^n c^n | n \geq 1\}$ ».

Якщо правильна – відбувається перехід на наступний крок.

**Крок 22.** Відбувається перехід до вікна з повідомленням про закінчення роботи та кнопкою повтору.

### 3.3 Блок-схема алгоритму тренажера

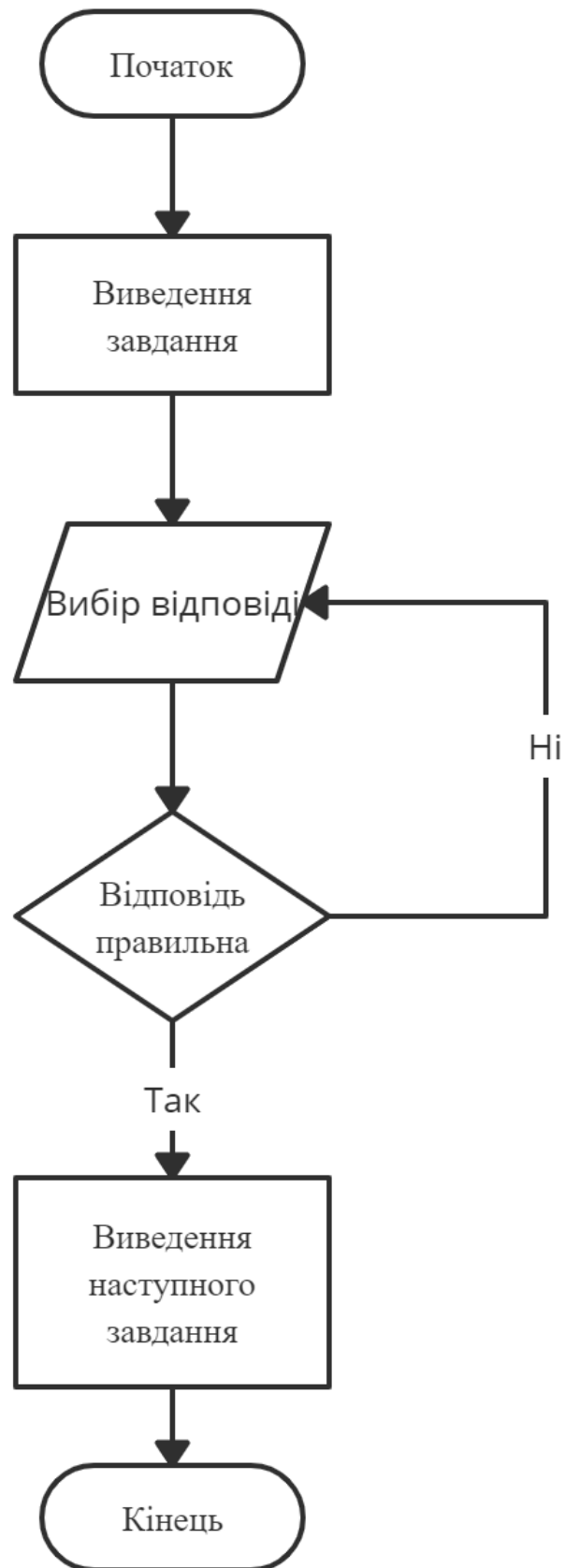


Рисунок 3.1 – Блок-схема роботи тренажера

## РОЗДІЛ 4. ПРАКТИЧНА ЧАСТИНА

### 4.1 Опис створення тренажера

Для початку роботи необхідно створити проект та задати в ньому певну ієрархію.

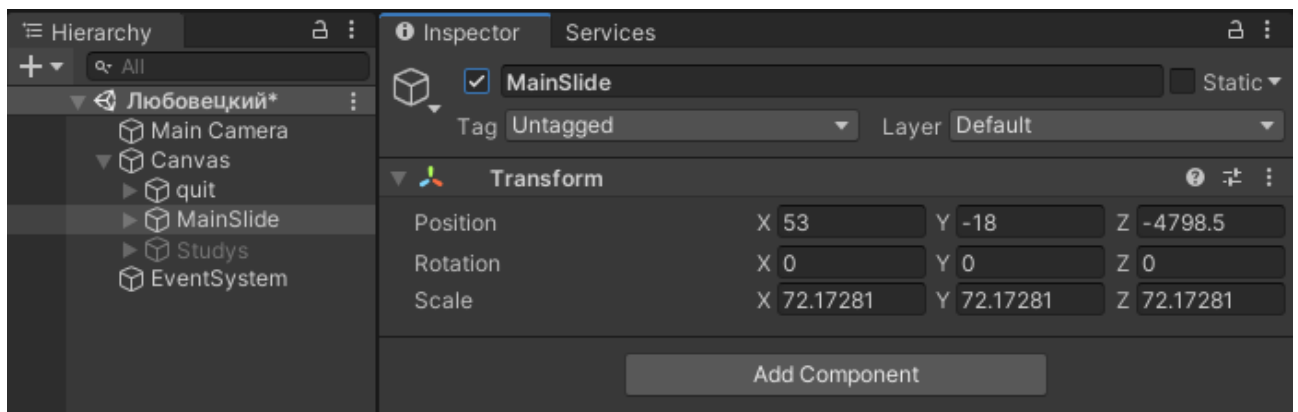


Рисунок 4.1 – Вікно ієрархії в Unity

Після створення ієрархії в проекті необхідно задати положення елементів у вікні проекту за допомогою меню навігації.

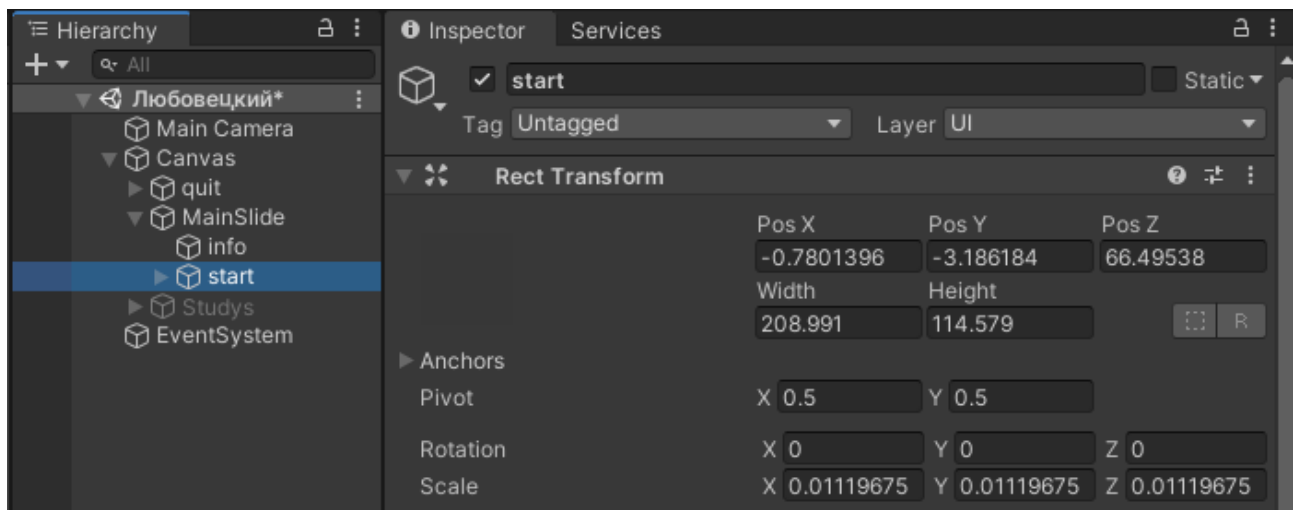


Рисунок 4.2 – Задання положення основних елементів тренажера

Навігацію між елементами тренажера виконано за допомогою скрипту для кнопок:

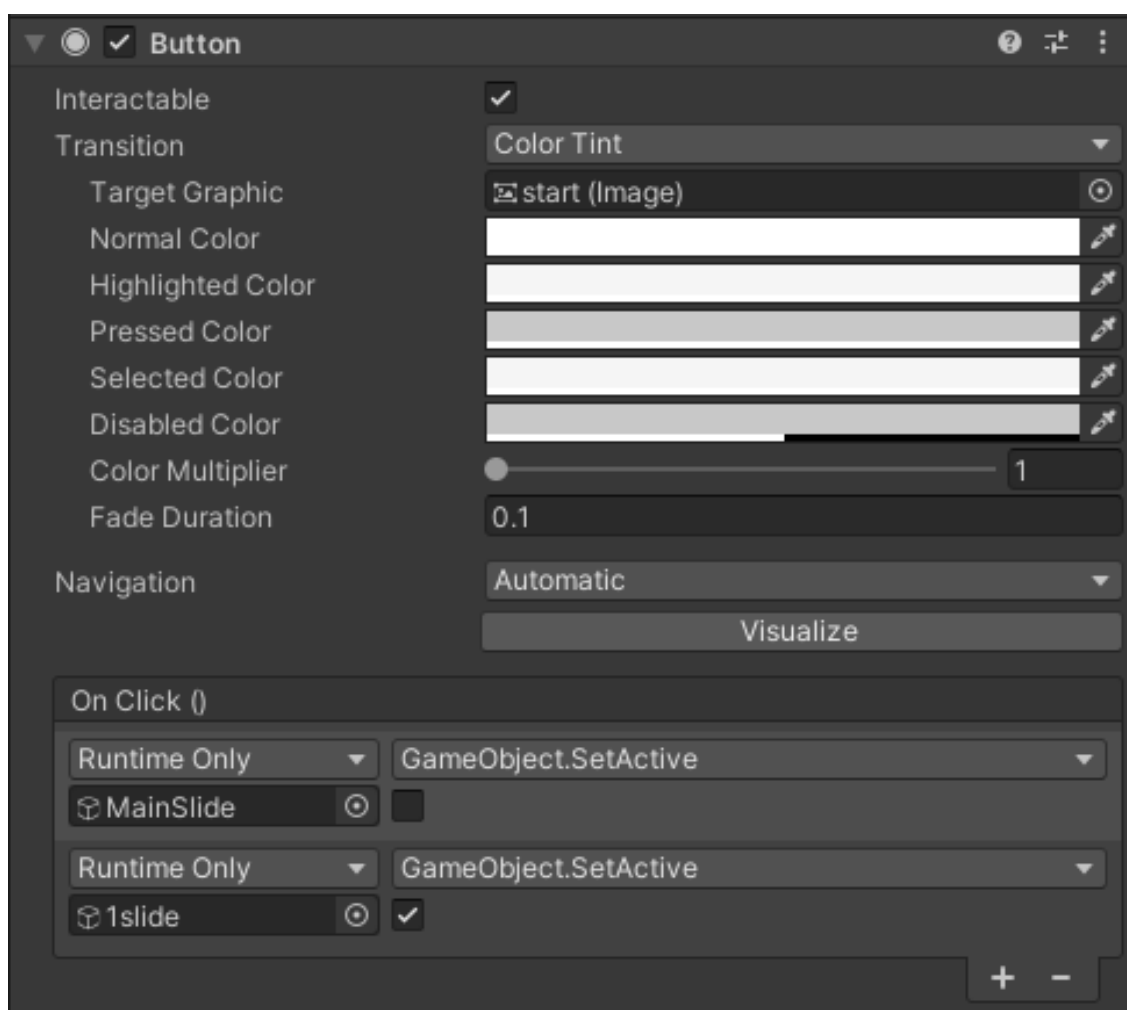


Рисунок 4.3 – Скрипт для кнопок

Після створення та задання положення в проєкті його основних елементів, а саме кнопок та текстових елементів, було створено ієрархію для тестових завдань.

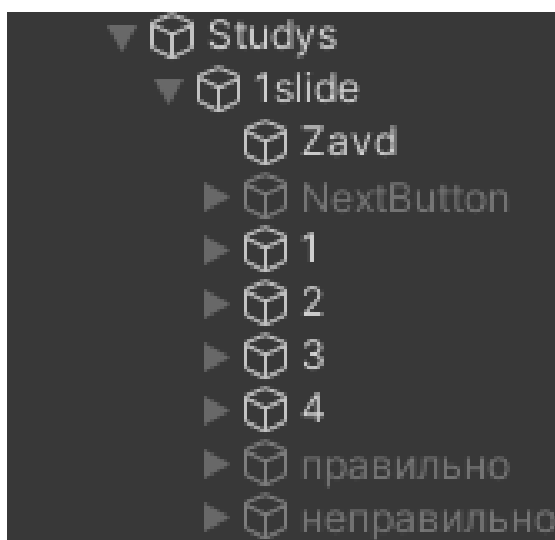


Рисунок 4.4 – Ієрархія для тестових завдань

Робота з текстом реалізована через скрипти для роботи з текстом.

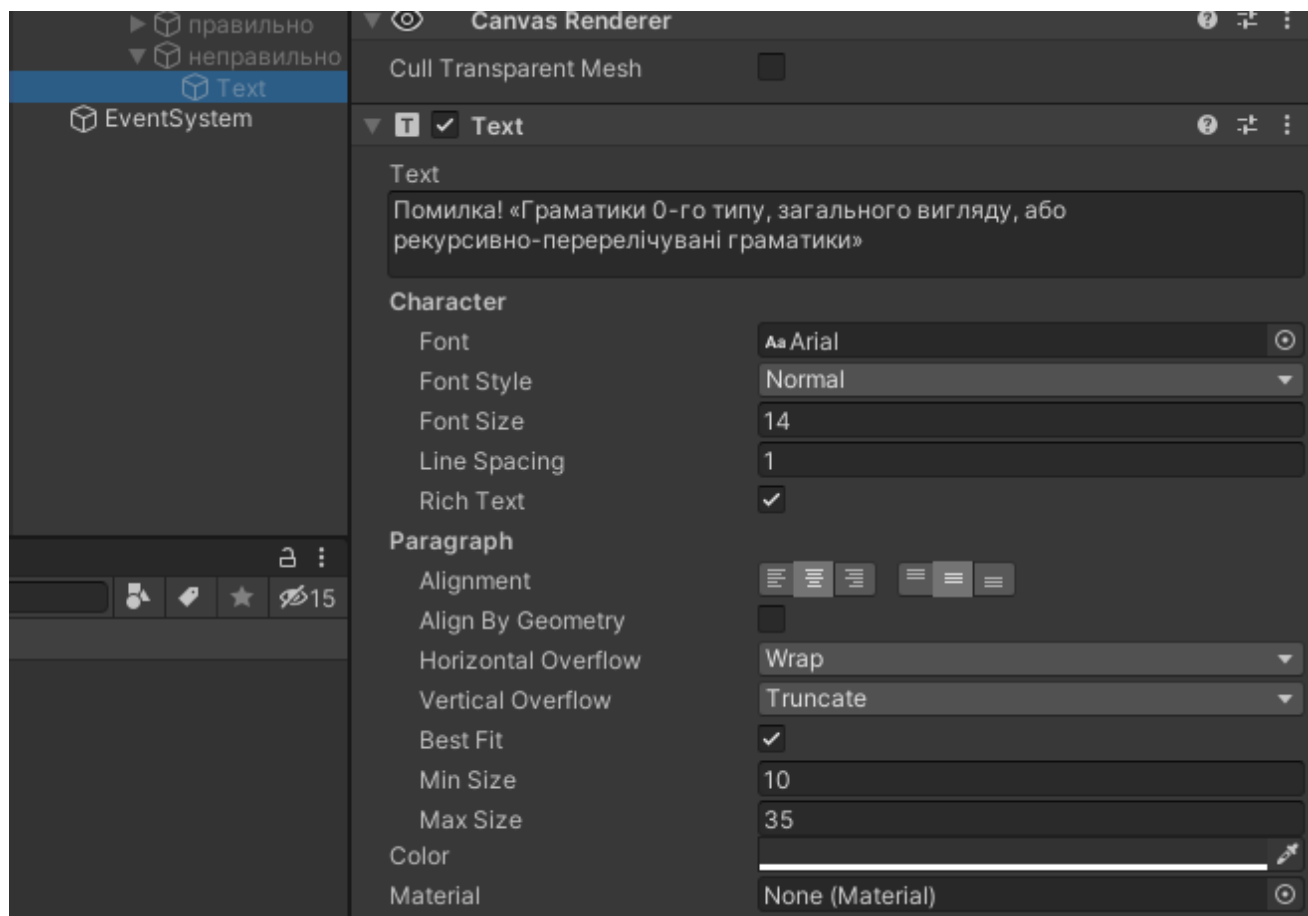


Рисунок 4.5 – Скрипти для роботи з текстом

Скрипт для роботи з кнопками зображено на рисунку 4.6:

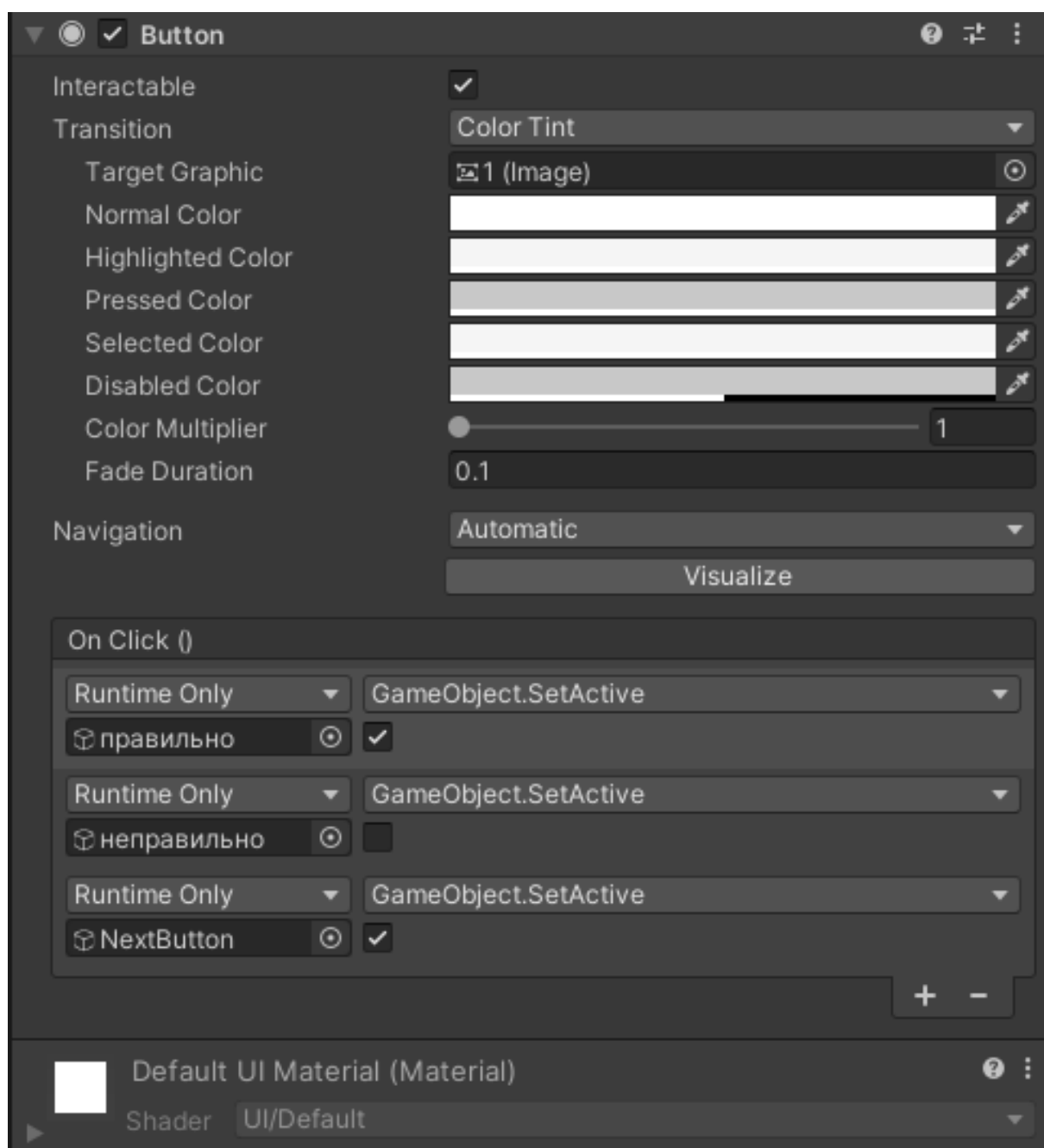


Рисунок 4.6 – Скрипт для роботи з кнопками

Після розташування всіх елементів та задання їм певних скриптів вікно тренажеру у вікні проекту виглядає наступним чином:



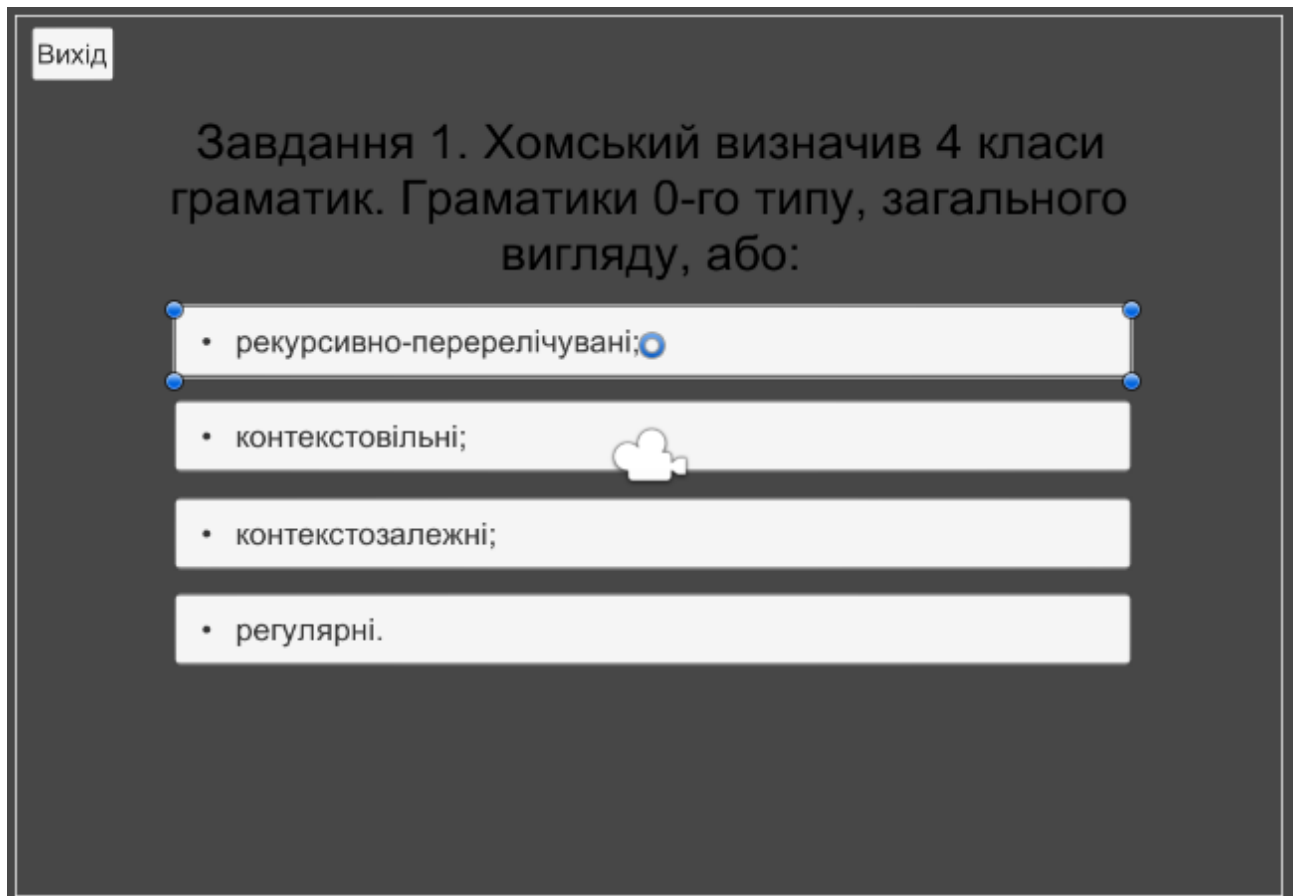


Рисунок 4.7 – Вікно з тестовим завданням в Unity

Після вибору правильної відповіді користувач отримує повідомлення та доступ до кнопки «Далі».

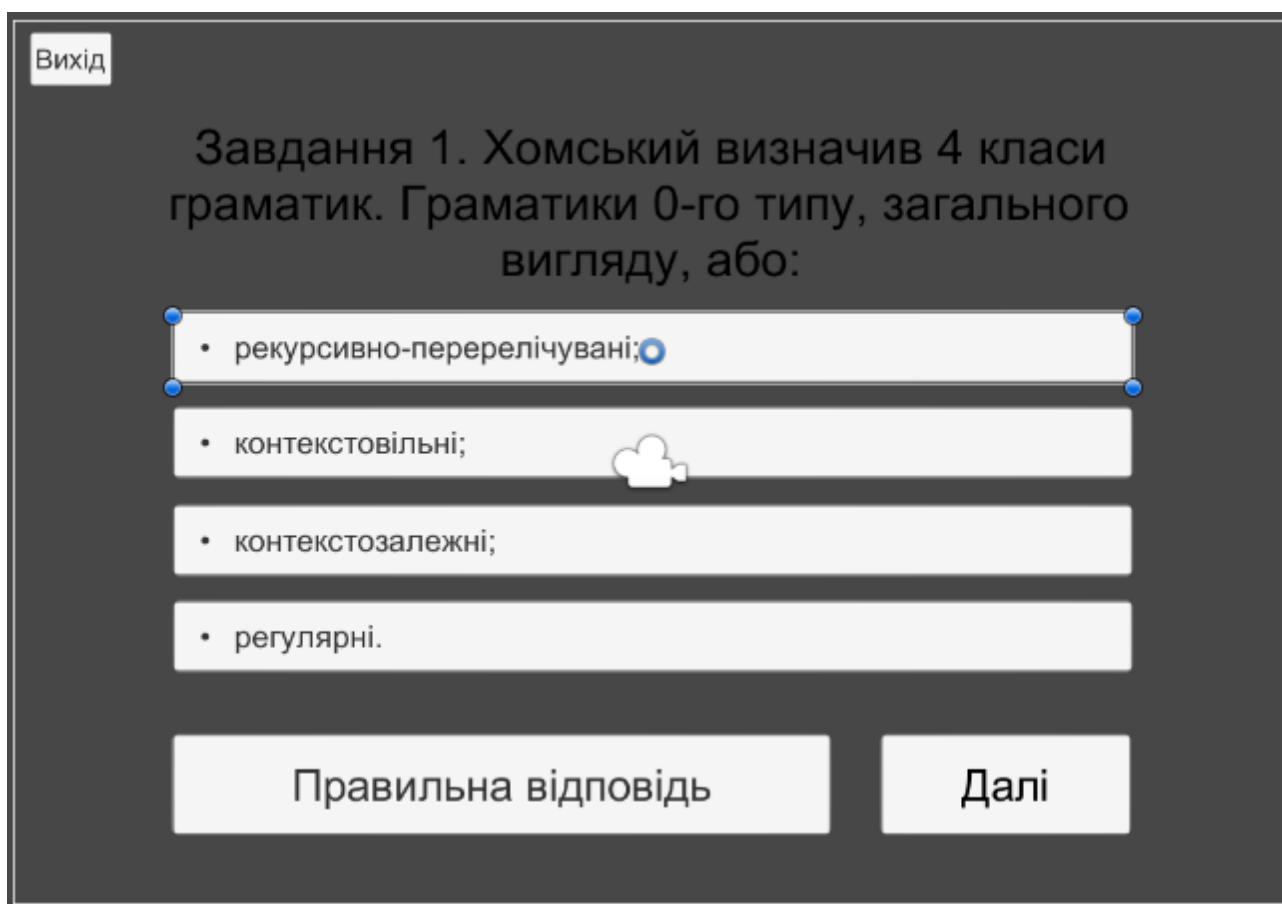


Рисунок 4.8 – Вікно з тестовим завданням в Unity після вибору правильної відповіді

Після вибору неправильної відповіді користувач отримує повідомлення з підказкою, та можливість обрати правильний варіант відповіді для продовження роботи з тренажером.

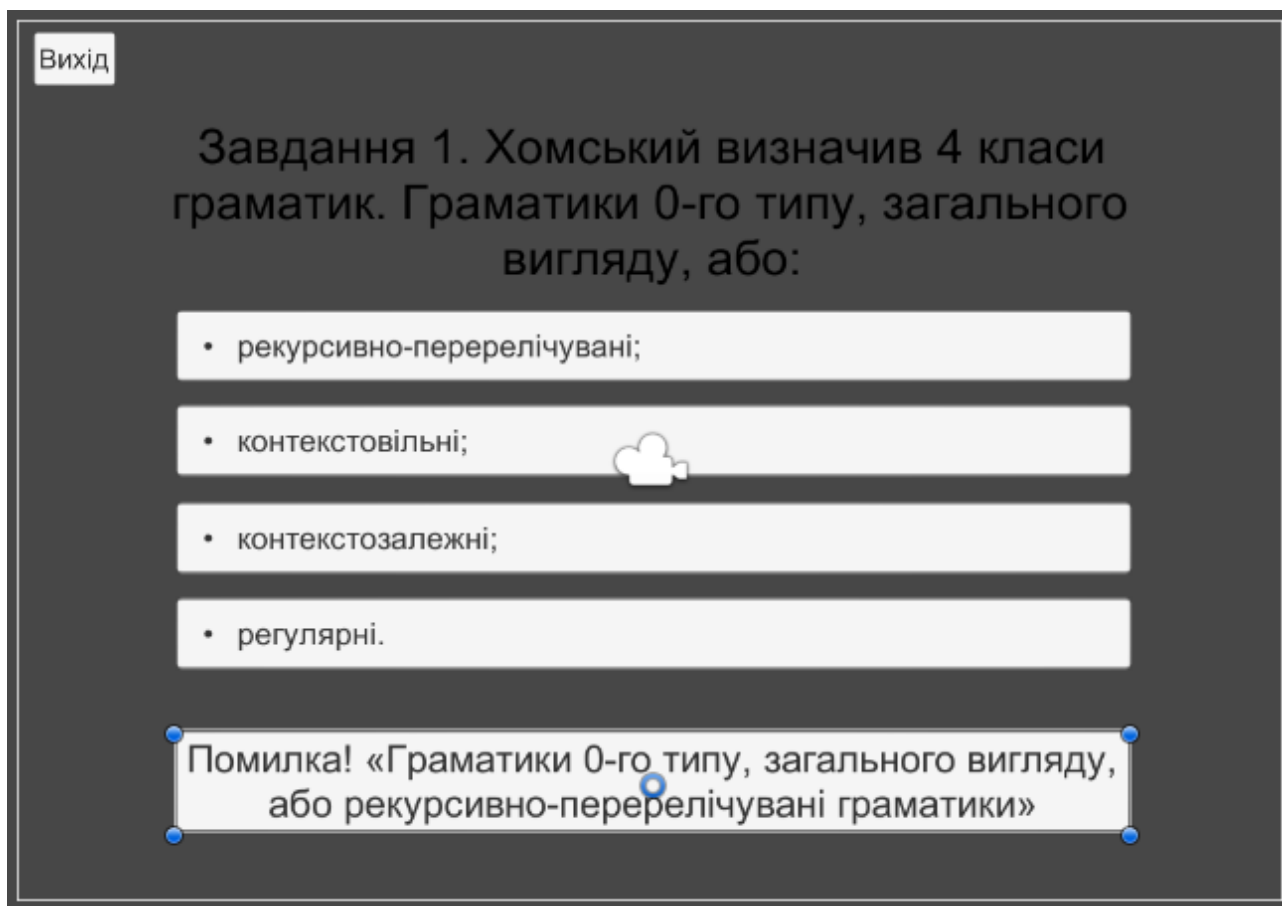


Рисунок 4.9 - Вікно з тестовим завданням в Unity після вибору неправильної відповіді

## 4.2 Інструкція для роботи з тренажером

Після запуску тренажеру користувач переходить до головного вікна з інформацією, що містить:

- тему;
- інформацію про автора;
- інформацію про керівника;
- перехід до виконання тренажеру;
- вихід.

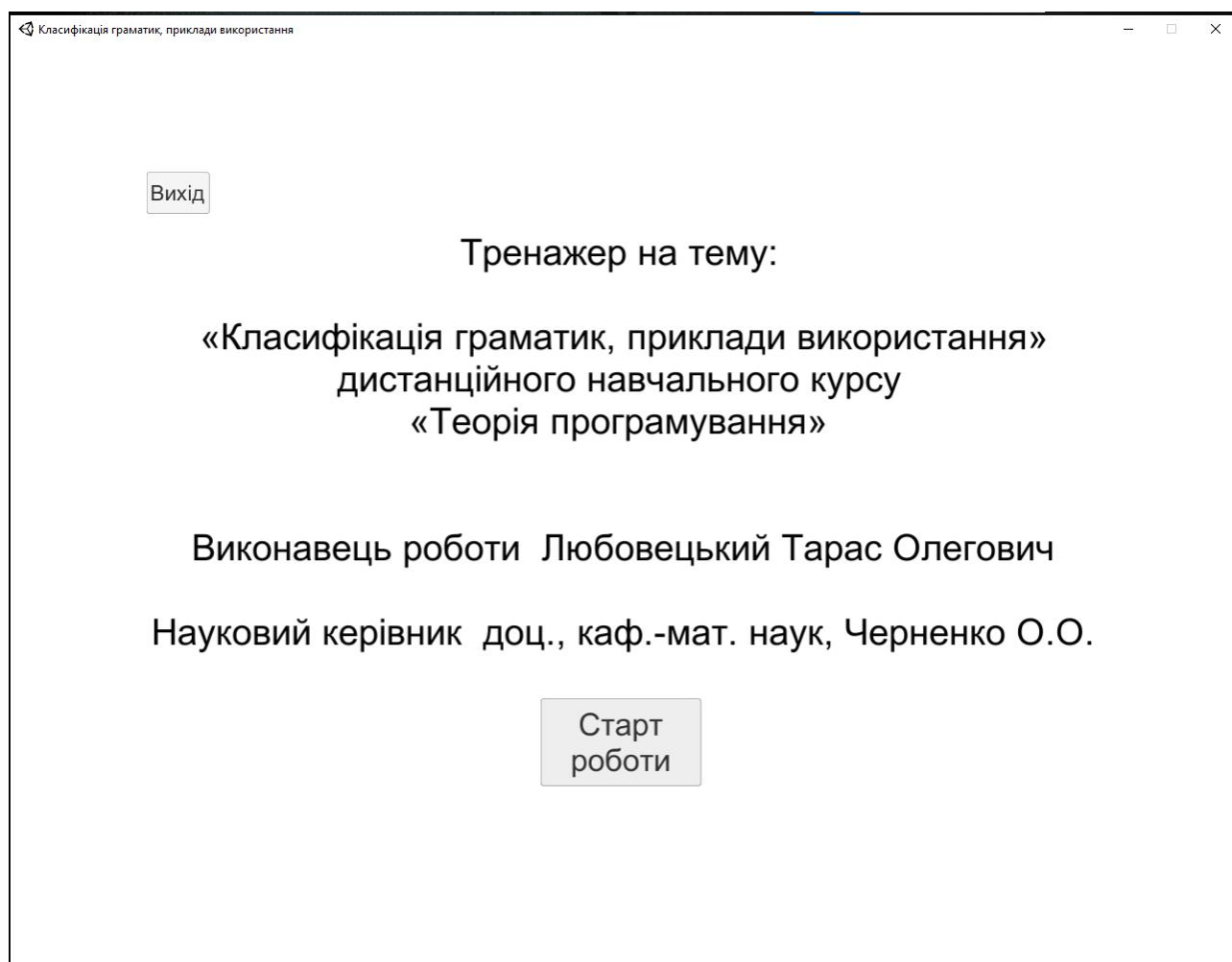
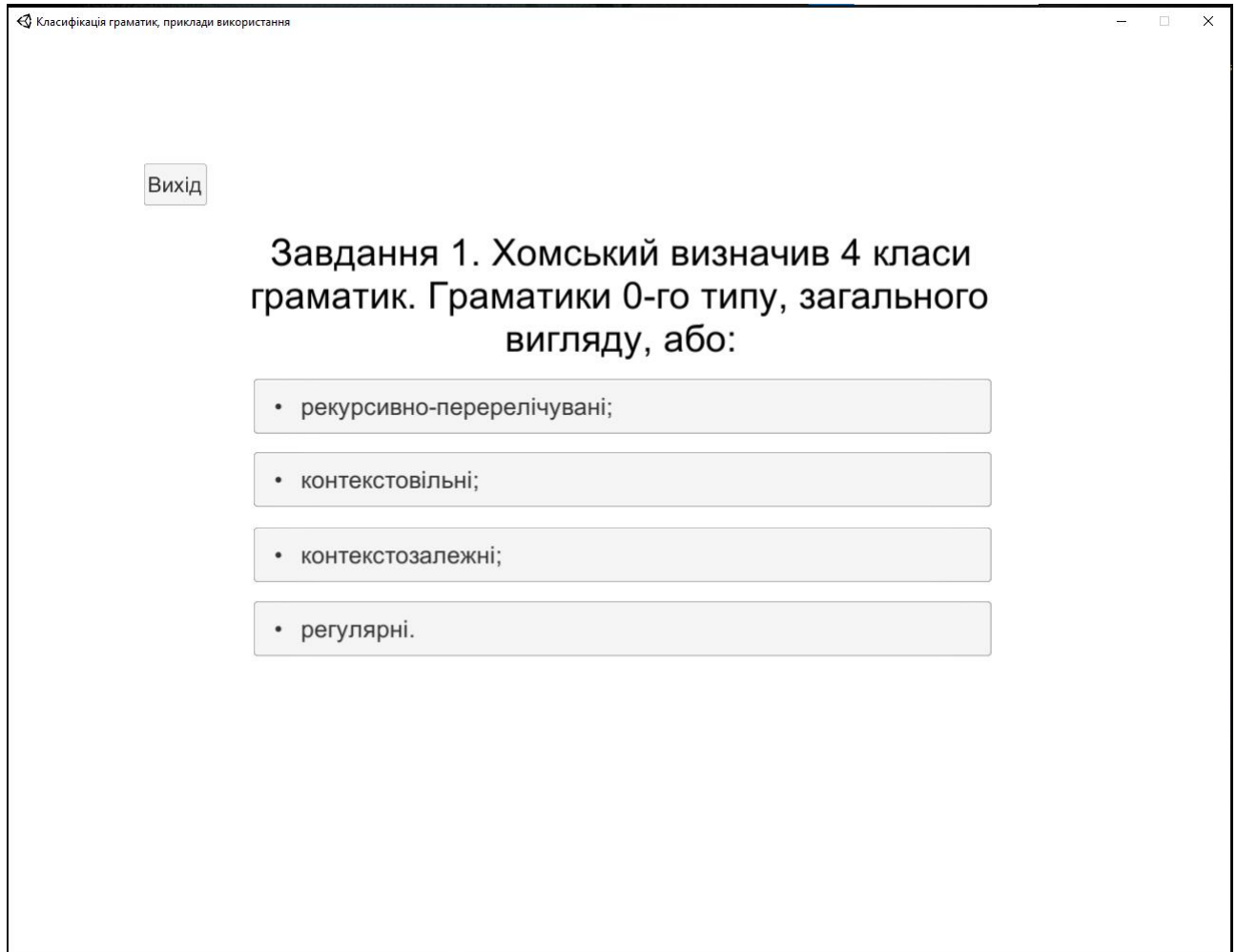


Рисунок 4.10 – Головне вікно тренажеру

Після натиснення на кнопку «Старт роботи» відбувається перехід до практичної частини тренажеру.

На кожному кроці виводиться завдання, необхідно вибрати відповідь.



Класифікація ґраматик, приклади використання

Вихід

Завдання 1. Хомський визначив 4 класи ґраматик. Ґраматики 0-го типу, загального вигляду, або:

- рекурсивно-перерелічувані;
- контекстовільні;
- контекстозалежні;
- регулярні.

Рисунок 4.11 – Зовнішній вигляд практичного завдання

Якщо відповідь неправильна, то відображається помилка: «Граматика 0-го типу, загального вигляду, або рекурсивно-перерелічувані граматики».

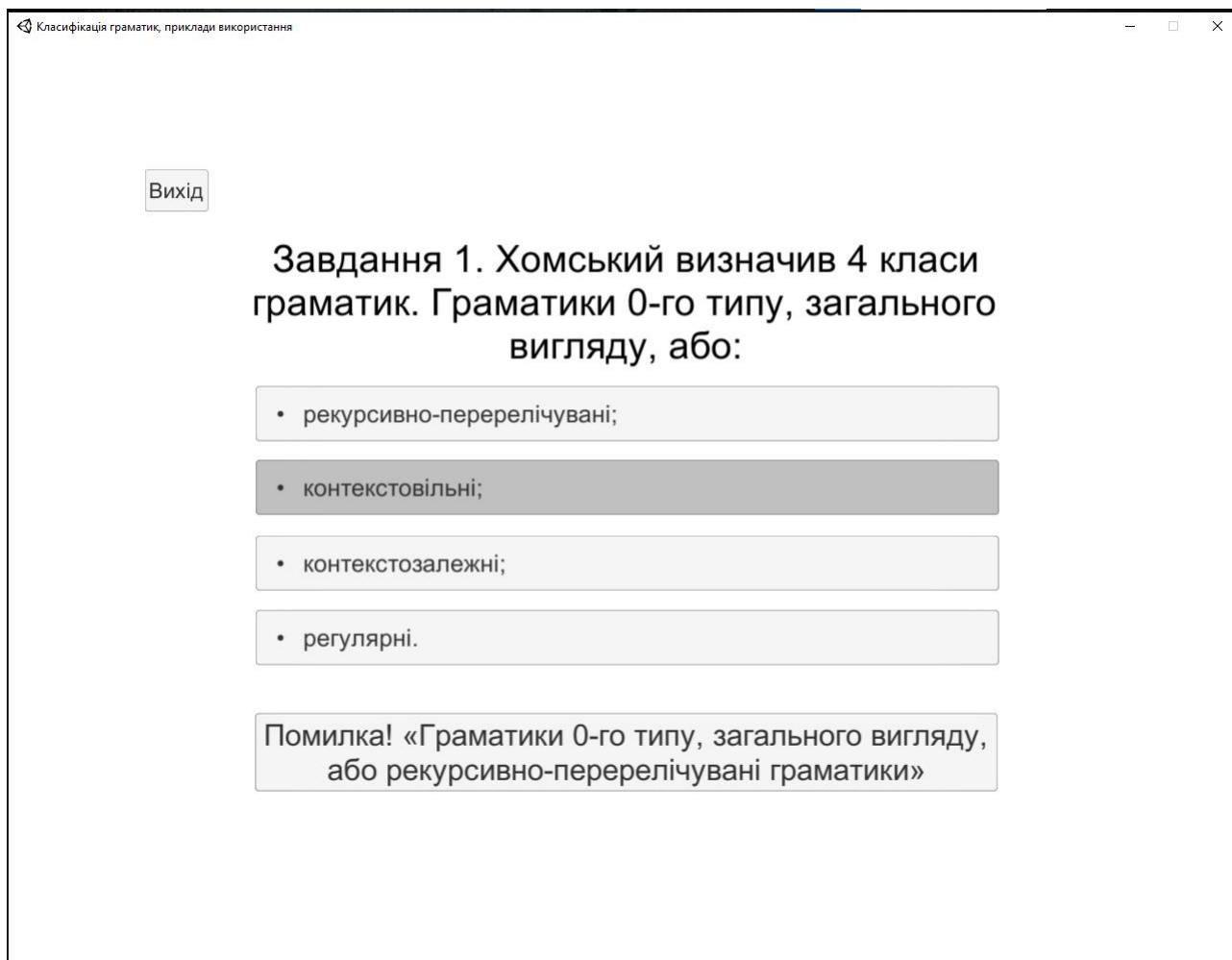


Рисунок 4.12 – Практичне завдання після вибору неправильної відповіді

Вікно тренажеру після вибору правильної відповіді.

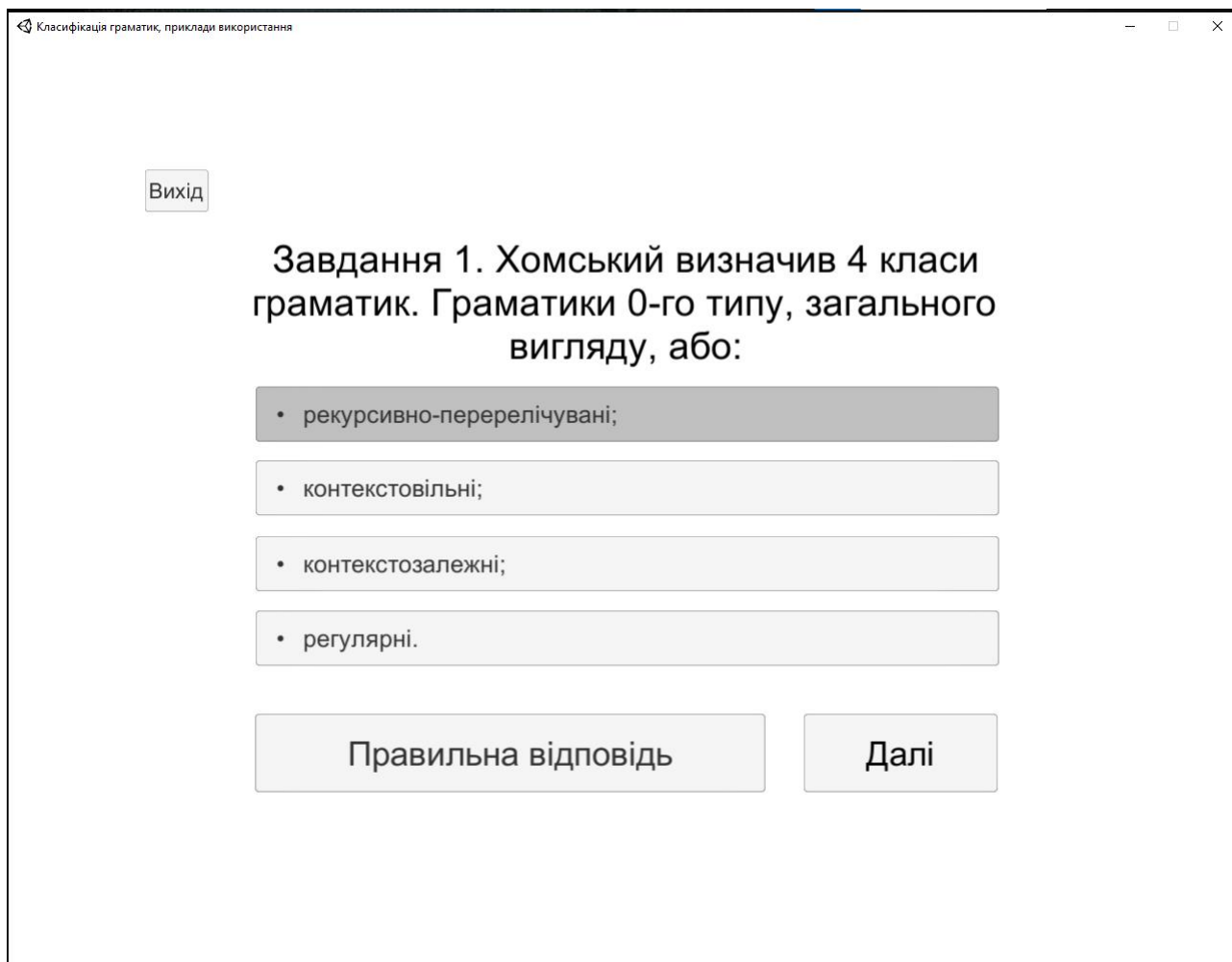


Рисунок 4.13 – Практичне завдання після вибору правильної відповіді





Якщо відповідь неправильна, то відображається помилка:

Класифікація граматики, приклади використання

Вихід

Завдання 13. Ієрархія Хомського є включаючою.  
Вкажіть по зображенню кожен тип:

Тип

Тип

)

Тип

)

Тип

)

Тип 3

Тип 2

)

Тип 1

)

Тип 0

)

Тип 2

Тип 2

)

Тип 0

)

Тип 1

)

Помилка! Оберіть перший варіант!

Рисунок 4.15 – Практичне завдання тестового типу з відповідями у вигляді зображень після вибору неправильної відповіді

Якщо правильна – відбувається перехід на наступний крок.

Класифікація граматик, приклади використання

Вихід

Завдання 13. Ієрархія Хомського є включаючою.  
Вкажіть по зображенню кожен тип:

Тип

Тип

)

Тип

)

Тип

)

Тип 3

Тип 2

)

Тип 1

)

Тип 0

)

Тип 2

Тип 2

)

Тип 0

)

Тип 1

)

Правильна відповідь

Далі

Рисунок 4.16 – Практичне завдання тестового типу з відповідями у вигляді зображень після вибору правильної відповіді

Після закінчення роботи з тренажером відбувається перехід до вікна з повідомленням про закінчення роботи та кнопкою повтору.

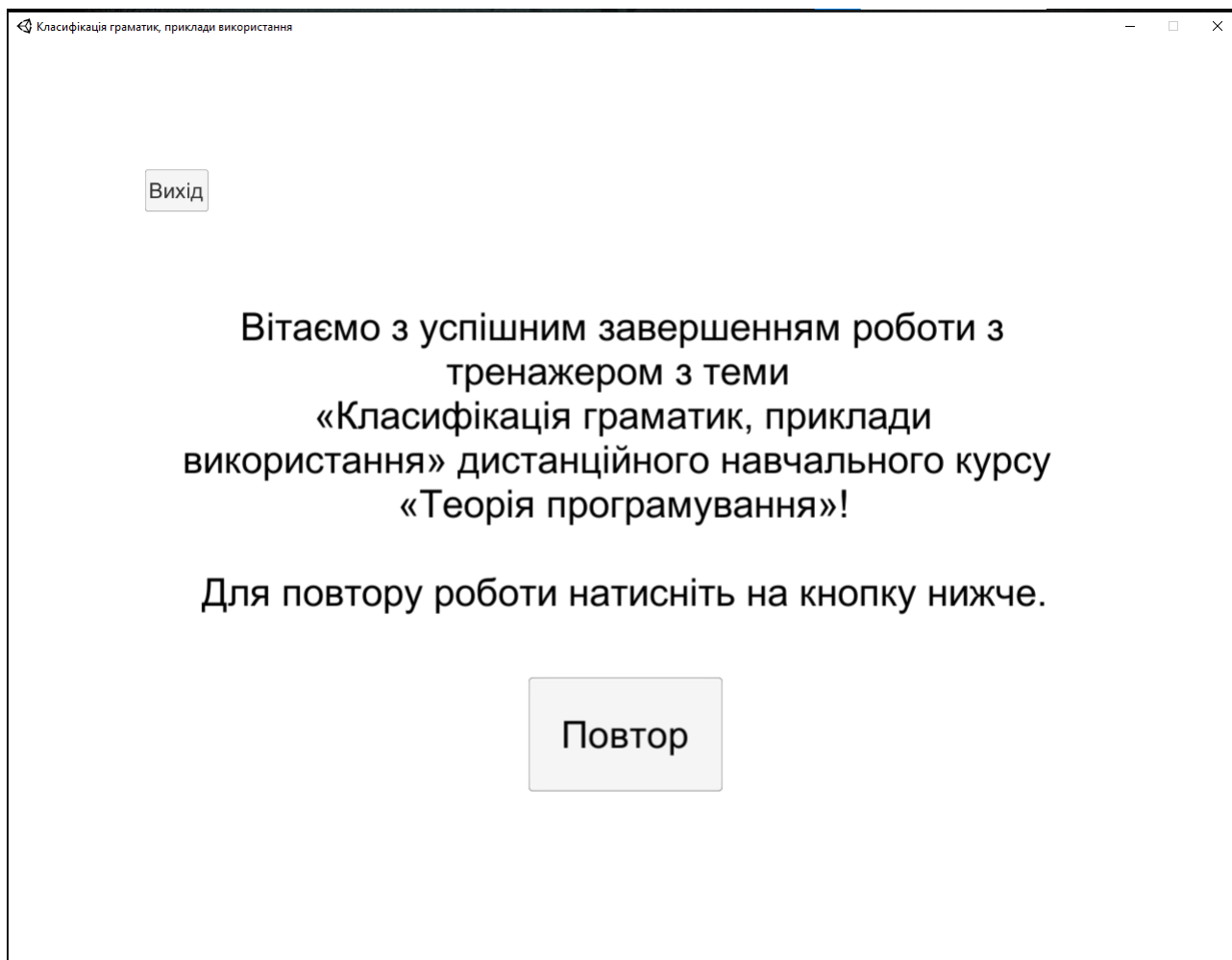


Рисунок 4.17 – Останній елемент тренажеру з кнопкою повтору

## ВИСНОВКИ

Головними завданнями або ж забезпеченням програм-тренажерів є:

- послідовне виведення на екран завдань заданої складності з вибраної теми;
- контроль за діями користувача з розв'язання запропонованого завдання;
- миттєву реакцію на неправильні дії;
- виправлення помилок користувача;
- демонстрацію правильного розв'язання завдання;
- виведення підсумкового повідомлення про результати роботи користувача (можливо, з рекомендаціями чи порадами).

Програма-тренажер повинна містити:

- стартову сторінку;
- сторінки з питаннями;
- повідомлення про помилку;
- сторінку з результатами проходження тренажера.

Для цього потрібно реалізувати наступні функції:

- перехід між основними панелями:
  - стартова сторінка;
  - сторінка з питаннями;
  - сторінка з результатами;
- перехід до наступного питання;
- перехід до попереднього питання;
- перевірка вибраної відповіді на питання;
- виведення повідомлення про помилку;
- підрахунок результатів проходження тренажера.

Широка інформатизація навчального процесу дозволяє створити умови, при яких відбувається розвиток креативних навичок і умінь, формується здатність аналізувати і прогнозувати завдання і процеси при вивченні різних

дисциплін. Саме дистанційні технології навчання дають можливість забезпечити студентів електронними навчальними ресурсами для проведення навчальної діяльності під час аудиторної роботи, служать ефективним інструментом для організації самостійної роботи, дозволяють реалізувати індивідуальний підхід до кожного студента.

Аналіз видів практичного заняття вказує на те, що вони є ефективними при організації дистанційного навчання при вивченні математичних дисциплін і сприяють якісному засвоєнню навчального матеріалу. Деякі з них є унікальними і їх реалізація можлива в інформаційному середовищі дистанційної платформи університету.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Ємець О. О. Методичні рекомендації щодо оформлення пояснювальних записок до курсових проектів (робіт) для студентів за освітньою програмою «Комп'ютерні науки» спеціальності 122 «Комп'ютерні науки та інформаційні технології», «Комп'ютерні науки» галузь знань - 12 «Інформаційні технології» / О. О. Ємець – Полтава : РВВ ПУЕТ, 2017. – 69с.
2. Білоус О.А. Дистанційний поточний контроль знань студентів при вивченні математичних дисциплін / О. А. Білоус, Ю. А. Кравченко – Актуальні питання природничо-математичної освіти. – 2016. – №7-8
3. Білоус О.А. Види практичної роботи при вивченні дистанційного математичного курсу / О. А. Білоус – Суми: Вид-во СумДУ
4. Черненко О.О. Електронний навчально-методичний посібник для самостійного вивчення навчальної дисципліни «Теорія програмування» для студентів напрямку 6.040302 «Інформатика»
5. Нікітченко М.С. Теоретичні основи програмування: Навчальний посібник [Електронний ресурс] / М.С. Нікітченко. – Київ: КНУ ім. Т.Г. Шевченка, 2009. – 200 с. – Режим доступу: <http://tp.unicyb.kiev.ua/doc/TOP.pdf>.
6. Бібліографічний запис. Бібліографічний опис. Загальні вимоги та правила складання: ДСТУ 7.1-2006. – [Чинний від 2007-07-01]. – К. : Держспоживстандарт України, 2007. – 47 с.
7. Майкл Фицджеральд, Регулярные выражения: основы: 2017, 240 стор.
8. Фридл, Дж. Регулярные выражения = Mastering Regular Expressions. — СПб.: «Питер», 2001. — 352 с.
9. Смит, Билл. Методы и алгоритмы вычислений на строках (regexr) = Computing Patterns in Strings. — М.: «Вильямс», 2006. — 496 с.
10. Платформа розробки Unity. – Режим доступу: <https://unity.com/ru>

## ДОДАТОК А

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine.UI;
using UnityEngine;

public class Inputs : MonoBehaviour
{
    public class QuitButton : MonoBehaviour
    {
        public void QuitGame()
        {
            Application.Quit();
        }
    }

    void Start()
    {
        Button btn = nxtButton.GetComponent<Button>();
        btn.onClick.AddListener(TaskOnClick);
    }

    void TaskOnClick()
    {
        Theme1.SetActive(false);
        Theme2.SetActive(true);
    }

    void Start()
    {
        Button btn = menuButton.GetComponent<Button>();
        btn.onClick.AddListener(TaskOnClick);
    }

    void TaskOnClick()

```

```
{
    Theme.SetActive(false);
    Menu.SetActive(true);
}
public void Next()
{
    Input.SetActive(false);
    AfterInput.SetActive(true);
    ShowInpTxt.text = inputdTxt.text;
}
}
```